Using the DAQ Control Line

Author: Jeff Landgraf

Components of the DAQ run control

To run DAQ with multiple sectors you need to have a front end program - either online's DAQ Server, or the DAQ control line program. In this document, the DAQ control line is described. The front end program must connect to the DAQ Run Control Handler, which handles the communication between the front end and the components of DAQ. Currently both the front end program and the handler must run on the computer *daqman.*

Starting the DAQ Run Control Handler

The Run Control Handler requires a file containing the DAQ hardware configuration. This file must exist before the Run Control Handler is started. To start the handler log into *daqman* as *daq.* Then, type:

```
cd /DAQ/bin/RC
handler –conf configfilename &
```

If you do not have a configuration file, see the section below for creating configuration files.

There is currently no message logging facility provided by DAQ. Messages are currently dumped to the file "/DAQ/log/msglog". To get a running list of the messages as they appear use the unix command:

```
tail –f /DAQ/LOG/msglog
```

Starting the DAQ command line:

Once the handler has been started you are ready to start the DAQ command line. Currently only one DAQ command line session should operate at one time. To start the session type:

```
cd /DAQ/bin/RC
daqrc
```

You should get the prompt "DAQ RC→." Press *return* to get a list of all of the valid commands.

Taking DATA

To take data perform the following steps:

1) Type "query" at the command line. The command should return "Current State: Ready". If not proceed to the section "Fixing problems"
2) Type "startrun" at the command line. You will be asked to supply 3 values. For each value a default will be indicated. If you wish to accept the default value, simply press *return*. If at any time you decide not to start the run type "cancel":
   a) Enter the run number.
   b) Enter what kind of run you want:
      i) Pedestal Run –

(1) Runs pedestal calculating code for all "normal" triggers
(2) Uses "seesaw" gain setting
(3) Pedestal subtraction is off
(4) Bypasses cluster finding
(5) Writes Raw data format by default
(6) Writes Pedestal data in the last event

ii) Gain Run -
(1) //Runs Cluster Finder for "normal" events
(2) Uses "logarithmic" gain setting
(3) Pedestal Subtraction is on
(4) Writes GAIN data by default

iii) Physics Run –
(1) //Runs Cluster Finder for "normal" events
(2) Uses "Gain Corrected Logarithmic" gain table // currently equivalent to "logorithmic"
(3) Pedestal Subtraction is set on
(4) Writes zero suppressed data format by default

iv) Test Run –
(1) This is only for use by "DAQ experts" who know how to use it.
(2) Run type, pedestal mode & gain mode are taken from the "Set" values for these parameters

c) Enter a number describing the destination for the data.

3) DAQ is now ready to start accepting triggers. The data will be placed in a file
a) If the EVD_DEST_MAIN variable is set to "ETHERNET" the file will be: "/data/scratch/evbData/taper1.R0##########" where # is the run number
b) If the EVB_DEST_MAIN variable is set to "SCSI" the file will be stored in: /DATA3/closed

4) To stop the run type "stoprun"
a) Currently, the program will display, "Error stopping run: There are still –1 tokens in the token manager. Do you still want to stop the run?" This is normal. Answer "y".

Setting Other Parameters

Additional run parameters are available using the "set" command . Type "set" at the command line. This command will return a list of variables, with their current values indicated. To change a value type "set *variable_name* value". Only change these variables if you already understand what they mean.

Fixing Problems

DAQ Run Control Handler Arguments

Most of the handler's command line arguments are used when running multiple instances of the handler for testing purposes. They should not be necessary for running the actual experiment. Here is the full list of command line parameters:

```
-nologtofile  - tells handler not to log to a file
-log filename - tells handler filename for logging
-debug #      - -1 no debug, each # * 1200 msgs
-debugtofile filnam - default daq.debug
-quiet        - only log CMD_LOG_DATA messages
-lport #      - give the listening port number
-padd #       - add # to both listen and send ports
-psuf chars   - suffix for pipes and log file
```

```
-cdir         - director for config file
-conf filnam  - filename for config file
-help         - this message
```

DAQ Run Control Commands

startrun – Start the run

stoprun – Stop the run

pauserun – Pause the run

resumerun – Resume the run

query – Get the current state of DAQ and write to the log file

trigger trigger_cmd #bunches <# in a bunch> - Send software triggers to DAQ

set <timeout> <variable> <value> - Set a variable or timeout

reconfig *filename* – Change the configuration filename

sendconfig – Sends the current configuration to DAQ components, but does not start the run

querytokens – Ask all of the DAQ components how many tokens they are currently processing. The value returned is the token managers value. The values for other components are displayed by the query command

cleardaq – Causes all of the DAQ components to flush their buffers

clearerror <state>– Changes the handlers understanding of all DAQ components state to the specified state. The default state is "Ready". You need to use the numeric state value for this command.

reboot – Reboot all of the connected DAQ components

stophandler – Exit run control and stop the handler

quit – Exit run control, but do not stop the handler

The following two commands allow you to control which components described in the configuration file are active in the run.

add *daqpath <rb_mask>* – Add a component to the run

subtract – Remove a component from the run

Example DAQ RC Hardware Configuration File

A sample configuration file is shown below:

```
begin: experiment STAR.1
# TPC Subsystem
begin: system TPC 1
```

```
begin: detector TPC
#
begin: crate tsb12
boards: in 1 2 3 4 5 6 7 8 9 10 11 12
ip:   tsb12
end: // crate tsb12
#
begin: crate evb01
ip: evb01
evb: in
end: // crate evb01
#
begin: crate gl3
ip: in gl3
gb:
end: // crate gl3
#
end: // Detector TPC
end: // System TPC
end: // Experiment STAR.1
```

This configuration file states that there are 3 DAQ entities in the system connected to an ethernet connection: the sector broker SB12, the event builder EVB01, and the global broker GL3. It also states that the sector broker SB12 controls 12 receiver boards.